Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

The INSC Security Infrastructure

S. Zeber

Defence R&D Canada – Ottawa
TECHNICAL REPORT
DRDC Ottawa TR 2004-156
December 2004

Canada

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
DEC 2004		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
The INSC Security Infrastructure (U)	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defence R&D Canada Ottawa,3701 Carling Avenue,Ottawa,CA,K1A 0Z4**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**76**	

The INSC Security Infrastructure

S. Zeber
DRDC Ottawa

Defence R&D Canada - Ottawa

Technical Report

DRDC Ottawa TR 2004-156

December 2004

Abstract

The INSC project was an international collaborative research and development activity established under an MOU among eight NATO nations to investigate and demonstrate a secure IPv6 network infrastructure capable of supporting a multi-national military coalition operation. The NATO C3 Agency also accepted an invitation to contribute although they did not sign the MOU. The goal was to demonstrate a network infrastructure that supported security, interoperability, manageability, and mobility. Security was provided at the Network Layer using the IPsec protocol. No security was provided at the Application Layer. The project successfully demonstrated a working security infrastructure with electronic key management to support the operation of IPsec devices. The main lesson resulting from this demonstration was that, although IPsec/IPv6 and electronic key management are both available independently as commercial technologies, they have not yet been fully integrated in a manner that provides a secure, easily manageable, scalable security infrastructure.

Résumé

Le projet de Réseaux Interopérables pour les Communications Sécurisées (RICS) était une activité de collaboration internationale de recherches et de développement établie sous un mémorandum d'accord parmi huit nations de l'OTAN pour étudier et démontrer une infrastructure sécurisée d'un réseau IPv6 capable de soutenir une opération militaire multinationale en coalition. L'agence C3 de l'OTAN a également accepté une invitation à contribuer bien qu'elle n'a pas signé le mémorandum d'accord. Le but était de démontrer une infrastructure de réseau qui soutient la sécurité, l'interopérabilité, la maintenance, et la mobilité. La sécurité a été fournie à la couche réseau en utilisant le protocole d'IPsec. Aucune sécurité n'a été fournie à la couche application. Le projet a avec succès démontré une infrastructure fonctionnante de sécurité avec la gestion électronique de clés pour soutenir le fonctionnement des dispositifs d'IPsec. La leçon principale résultant de cette démonstration était que, bien que IPsec/IPv6 et gestion électronique des clés soient les deux disponibles indépendamment en tant que technologies commerciales, ils n'ont pas encore été entièrement intégrées pour fournir une infrastructure sécurisée, facilement maniable ou même à capacité d'expansion.

This page intentionally left blank.

Executive summary

The INSC project was an international collaborative research and development activity established under an MOU among eight NATO nations (CA, FR, DE, IT, NL. NO, UK, US) to investigate and demonstrate a secure IPv6 network infrastructure capable of supporting a multi-national combined joint task force coalition operation. The NATO C3 Agency also accepted an invitation to contribute, although they did not sign the MOU. The goal of the project was to demonstrate an infrastructure that supported security, interoperability, manageability and mobility. Security was provided at the Network Layer using the IPsec protocol. No security was provided at the Application Layer.

The project implemented and successfully demonstrated a network architecture and a working security infrastructure in which national and coalition local area network enclaves connected through IPsec devices to unprotected, or black, wide area network services. Coalition communications between users and applications in different coalition LANs took place over virtual private network connections tunnelled through the black domain transit networks, which could include national security domains as well as unclassified public domains. IP packets in the VPN tunnels were protected by IPsec encryption.

The main problem of the security infrastructure was to provide trusted scalable key management for the cryptographic keys used by the IPsec devices. Initial testing was performed using manual key distribution but a scalable solution required electronic key management. A PKI was implemented using open source software to support the distribution and management of signature keys for IPsec device authentication. The session keys used to encrypt IP packets were established over a connection between authenticated devices.

This report describes the key management infrastructure that was implemented, analyses its properties and notes the lessons learned from this project. The main findings resulting from the implementation and demonstration of the security infrastructure are noted below.

IPsec/IPv6 implementations are largely available both as open source and commercial product implementations and are interoperable. However the integration with electronic key management ranges from immature to non-existent. Certain IPsec implementations include a limited capability for X.509-based authentication but full support for automated key management services including dynamic certificate and CRL management remains to be achieved. Additionally, commercial PKI and X.500 directory products mostly do not yet support IPv6. No IPv6-enabled commercial products could be found for INSC.

Security policy and procedures were adopted pragmatically as required for operational effectiveness, as was the trust model for key management, which amounted to a de-facto cross-certification agreement. However this experience provides a good basis for developing a security policy for a coalition environment.

Finally, although INSC successfully demonstrated a working security infrastructure using IPsec with electronic key management to secure communications in a coalition network, full COTS integration of IPsec with electronic key management remains to be achieved.

Zeber, S. 2004. The INSC Security Infrastructure. DRDC Ottawa TR 2004-156. Defence R&D Canada - Ottawa.

Sommaire

Le projet de Réseaux Interopérables pour les Communications Securisées (RICS) était une activité de collaboration internationale de recherches et de développement établie sous un mémorandum d'accord parmi huit nations de l'OTAN pour étudier et démontrer une infrastructure sécurisée d'un réseau IPv6 capable de soutenir une opération militaire multinationale en coalition. L'agence C3 de l'OTAN a également accepté une invitation à contribuer bien qu'elle n'a pas signé le mémorandum d'accord. Le but était de démontrer une infrastructure de réseau qui soutient la sécurité, l'interopérabilité, la maintenance, et la mobilité. La sécurité a été fournie à la couche réseau en utilisant le protocole d'IPsec. Aucune sécurité n'a été fournie à la couche application. Le projet a avec succès démontré une architecture de réseau et infrastructure de sécurité fonctionnante dans laquelle les systèmes de réseaux locaux nationaux et réseaux de coalition connectés via dispositifs IPsec à des réseaux non protégés, ou noir, aux services de réseaux larges.

Les communications en coalition entre les utilisateurs et les applications à travers le réseau local d'une coalition différente ont eu lieu au-dessus des raccordements de réseaux virtuels privés (RVP) acheminés à travers les domaines de réseaux noirs de transit, qui pourraient inclure des domaines de sécurité nationale aussi bien que des domaines publics non classifiés. Les paquets d'IP dans les tunnels de RVP ont été protégés par l'enchiffrage d'IPsec. Le problème principal de l'infrastructure de sécurité était de fournir une gestion des clés cryptographiques capable de s'accroître d'une façon commensure pour les dispositifs d'IPsec. L'essai initial a été réalisé en utilisant une distribution manuelle des clés mais la gestion d'un système capable de s'accroître exigeait une gestion électronique des clés. Une infrastructure à clé publique a été mise en application en utilisant un logiciel de source ouverte afin de soutenir la distribution et la gestion des clés de signature pour l'authentification des dispositifs d'IPsec. Les clés de session employées pour chiffrer des paquets d'IP ont été établies au-dessus d'un raccordement entre les dispositifs authentifiés.

Ce rapport décrit l'infrastructure de la gestion des clés qui a été mise en application, il analyse ses propriétés et note les leçons apprises de ce projet. Les résultats principaux résultant de l'exécution et de la démonstration de l'infrastructure de sécurité sont notés ci-dessous.

Les implementations IPsec/IPv6 sont en grande partie disponibles en tant que source ouverte ou produit commercial et sont interopérables. Cependant l'intégration avec la gestion électronique des clés s'étend de non mûr à inexistant. Certaines implementations d'IPsec incluent des possibilités limitées pour l'authentification de X.509 pour l'authentification, mais le support des services de gestion des clés automatisées, incluant certificat dynamique et la gestion de revocation de certificat demeure à être réalisé. En plus, les produits commerciaux de clés publiques et d'annuaire X.500 pour la plupart ne soutiennent pas encore IPv6. Aucun produit commercial d'IPv6 n'a pu être trouvé pour RICS.

Les règles et les procédures de sécurité ont été adoptées de façon pragmatique pour assurer l'efficacité opérationnelle et ce même pour ce qui est du lien de confiance pour la gestion des clés, qui est devenue l'accord ***de facto*** pour la cocertification . Cependant cette expérience

fournit une bonne base pour développer une politique de sécurité pour un environnement en coalition.

En conclusion, bien que RICS ait avec succès démontré une infrastructure fonctionnante de sécurité en utilisant IPsec avec la gestion électronique des clés pour fixer des communications dans un réseau de coalition, la pleine intégration de produits commerciaux IPsec avec la gestion électronique des clés demeure toutefois à être réalisée.

Zeber, S. 2004. The INSC Security Infrastructure. DRDC Ottawa TR 2004-156. R & D pour la défense Canada - Ottawa

Table of contents

List of figures

Acknowledgements

The author performed the work described in this report as a member of the Canadian INSC project team led by Dr. John Robinson, research director of the Network Systems and Technologies section at the Communications Research Centre.

The author would like to thank Mr. Vincent Taylor of DRDC Ottawa, and Mr. Tim Symchych of the Communications Research Centre for providing valuable comments on the initial draft of this report.

This page intentionally left blank.

1. Introduction

1.1 INSC Overview

The Interoperable Networks for Secure Communications (INSC) project was an international collaborative research and development activity established by a Memorandum of Understanding (MOU) among eight nations in the North Atlantic Treaty Organization (NATO): Canada, France, Germany, Italy, Norway, the Netherlands, the United Kingdom and the United States. The NATO Consultation, Command and Control Agency (NC3A) also accepted an invitation to contribute but did not sign the MOU.

The objective of the INSC project was to develop and demonstrate a common technical architecture for a multi-national military network comprising various military and civil subnetworks, including mobile networks, with the following properties:

- interoperable,
- secure,
- manageable, and
- based on existing and emerging standards, and commercial services and products.

A major interest was to examine the suitability of the Internet Protocol, version 6 (IPv6) and the Internet Security Protocol (IPsec) technologies, to support a multi-national military coalition environment and to demonstrate an evolutionary path from IPv4 to IPv6 for future coalition networks.

The technical work was divided among the following eight task areas, each with its own set of objectives and work plan, all overseen by an INSC Steering Committee:

- Task 1: System Architecture,
- Task 2: Information Services,
- Task 3: Network Management,
- Task 4: Security,
- Task 5: Routing,
- Task 6: Mobility,
- Task 7: Sub-networks, and
- Task 8: Directory Services.

The planned timescale of the INSC project called for demonstrations to be held and completed within three years of the completion of the MOU signing process. The work began in early 2001 and was completed by October 2003. National demonstrations were held from September to November 2003 and the project results were presented at an international symposium held at the NATO C3 Agency in The Hague in November 2003.

1.2 The INSC Scenario

Future military operations are foreseen to involve either a multi-national Combined Joint Task Force (CJTF) or a national Joint Task Force (JTF) consisting of land, sea and air forces. The operational scenario adopted as the context for the INSC project was a littoral warfare environment involving air, sea, and land forces, which is expected to be typical of future NATO and other coalition military operations. The scenario, illustrated in Figure 1, depicts a multinational CJTF arriving by sea, with air support, landing and establishing a land-based operation including mobile units, all the while maintaining secure communications among the land, sea and air units and among national enclaves. Each ship, aircraft, and mobile land unit in Figure 1 is considered to be an autonomous system (AS) for the purpose of communication.

The nature of the INSC project was to investigate the technologies required to maintain seamless secure communications among the various operational units, and the communications challenges imposed by the characteristics of this littoral warfare environment involving multinational air, sea, and land forces.

Figure 1. *INSC Operational Scenario*

1.3 About This Report

This report describes the INSC Security Infrastructure that was implemented and demonstrated from the perspective of Canadian national participation and Canadian contributions to this work in the Security and Directory tasks. The work was performed in the period from 2001-2003. For a complete and comprehensive description of the project and its results, the reader is referred to the official INSC reports [1], [2], [3], [4], [5].

Chapters 1 - 3 of this report provide an overview of the INSC project, the INSC network, and the INSC security architecture that was implemented. Chapters 4 - 6 describe the details of the security management infrastructure and Chapter 7 describes the Canadian national demonstration of security management. Finally, Chapter 8 presents observations on, and an analysis of the operation of the security infrastructure, and Chapter 9 presents the lessons learned and conclusions.

This report includes some information presented in other INSC reports in order to make the report as self-contained as possible.

2. The INSC Network

This chapter provides an overview of the INSC network. More extensive details can be found in [1], [2], [3], [4], and [5].

2.1 Design Principles

NATO doctrine on communications for coalition operations [6] requires that a coalition network should link the defence headquarters of the nations, the coalition commanders, and the tactical operational units, both coalition and national. The INSC network design therefore included the following operational components:

- a Joint Command (JC) component
- a Land (L) component, and
- a Maritime (M) component.

Each operational component (JC, L, M) comprised a wide area network (JCWAN, LWAN, MWAN) interconnecting national and/or coalition local area networks (CLANs) to support the operations of the corresponding military force component (Joint Command, Land or Sea). The JCWAN, the LWAN and the MWAN were considered to be separate interconnected Autonomous Systems. There was no explicit Air (i.e., no AWAN) component, although the architecture is easily extended to include such a component if required.

The following additional design principles [7] were also adopted for the INSC network:

- There is only one coalition.
- There are two coalition security domains, a red, or protected, domain, and a black, or unprotected, domain.
- All coalition LANs belong to the red domain and operate at the same level of classification.
- The red domain is separated from the black domain and protected by IPsec security mechanisms.
- Security services are provided only at the Network Layer using the IPsec mechanism.

The following implementation constraints were also adopted as a pragmatic decision in order to limit and simplify the scope of the implementation:

- There are no Application Layer security services or mechanisms.

- All INSC data is UNCLASSIFIED.
- INSC will use only publicly available cryptographic algorithms in IPsec.

The first constraint was due to the fact that none of the proposed applications included the necessary support to be able to use Application Layer security services. To include this feature would have required development and integration resources that exceeded the level resources committed to INSC. The second and third constraints were adopted to avoid the procedural overhead that a classified environment would have imposed on the project. The goal of INSC was technology investigation and demonstration, and not the development of an operational capability.

Finally, a naming and addressing plan [8] was developed for the INSC network that established common naming and network addressing conventions and assigned IPv6 addressing domains to each nation.

2.2 Operational Architecture

Figure 2 shows a typical operational view of the INSC network architecture, including the wide area networks (JCWAN, LWAN, and MWAN), CLANs and example national network and LAN components. The CLAN on the JCWAN could be an element of the joint command headquarters (JCHQ), while the shore stations (Nation A and Nation B) represent national elements supporting the coalition commander. CLANs on the LWAN represent tactical elements supporting land operations. A stationary CLAN may be associated with a fixed field headquarters, while roving mobile units may connect through the mobile network. A nation may also contribute national support through a national tactical network. Ships from participating nations linked by the MWAN provide maritime support. Each ship may include both national and coalition components.

This architecture provides complete wide area connectivity so that any user or system in any CLAN or national LAN may communicate with any other coalition or national user or system supporting the operation.

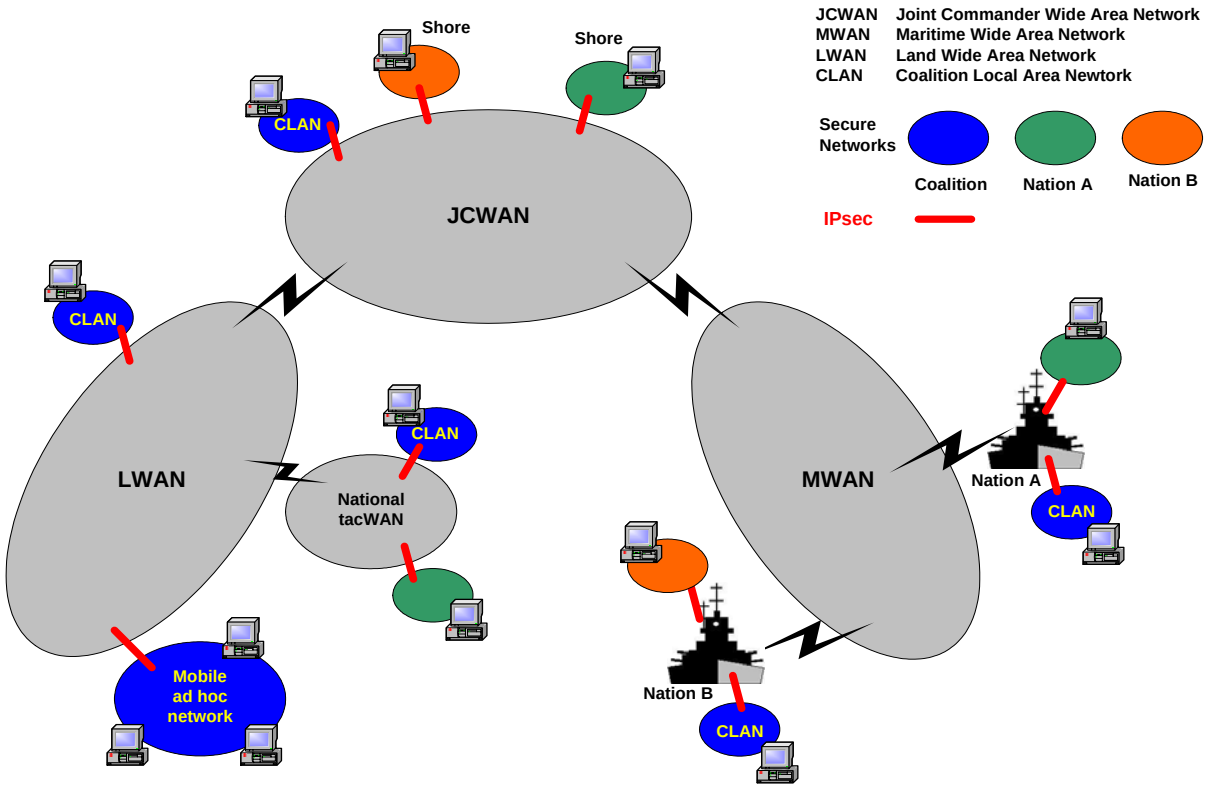

Figure 2. *INSC Network Architecture*

2.3 The Wide Area Networks

Each of the wide area networks was built by linking the routers of participating nations using the following bearer services:

- the 6bone (IPv6) [9],
- the Internet (IPv4), and
- commercial Integrated Services Digital Network (ISDN).

Each nation decided which bearer services it would support in each of the wide area networks. Figure 3 shows the international wide area network topology for each of the wide area networks.

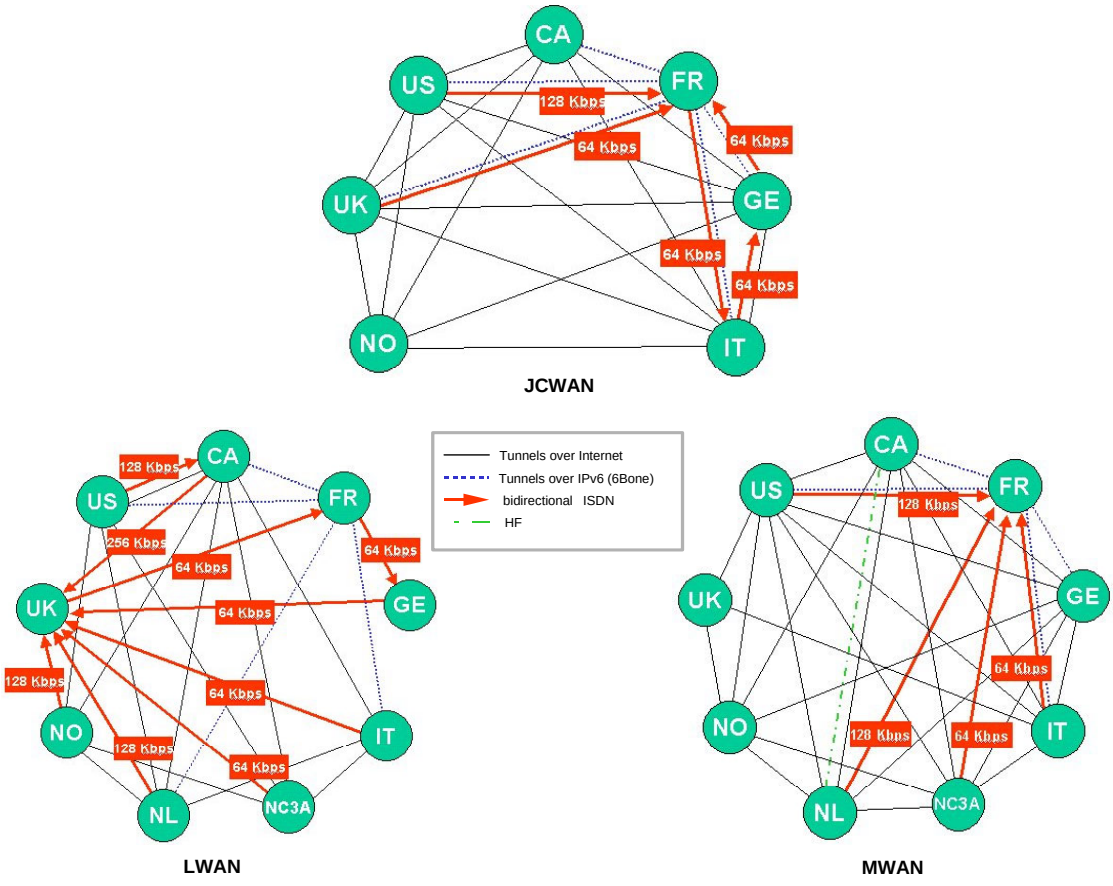

Figure 3. *INSC WAN Topology*

2.4 The Coalition Local Area Network

The coalition local area network in INSC hosts the applications and services that support the coalition operation. Figure 4 shows the structure of a typical CLAN. The applications can include email, web services, voice and video, and the red domain infrastructure services include routing, a domain name system (DNS) service and network management. The CLAN connects to the black domain wide area network router through an IPsec gateway device. Figure 4 also shows black domain directory, DNS and network management services located on the black side of the IPsec gateway, but on the CLAN side of the black domain router.

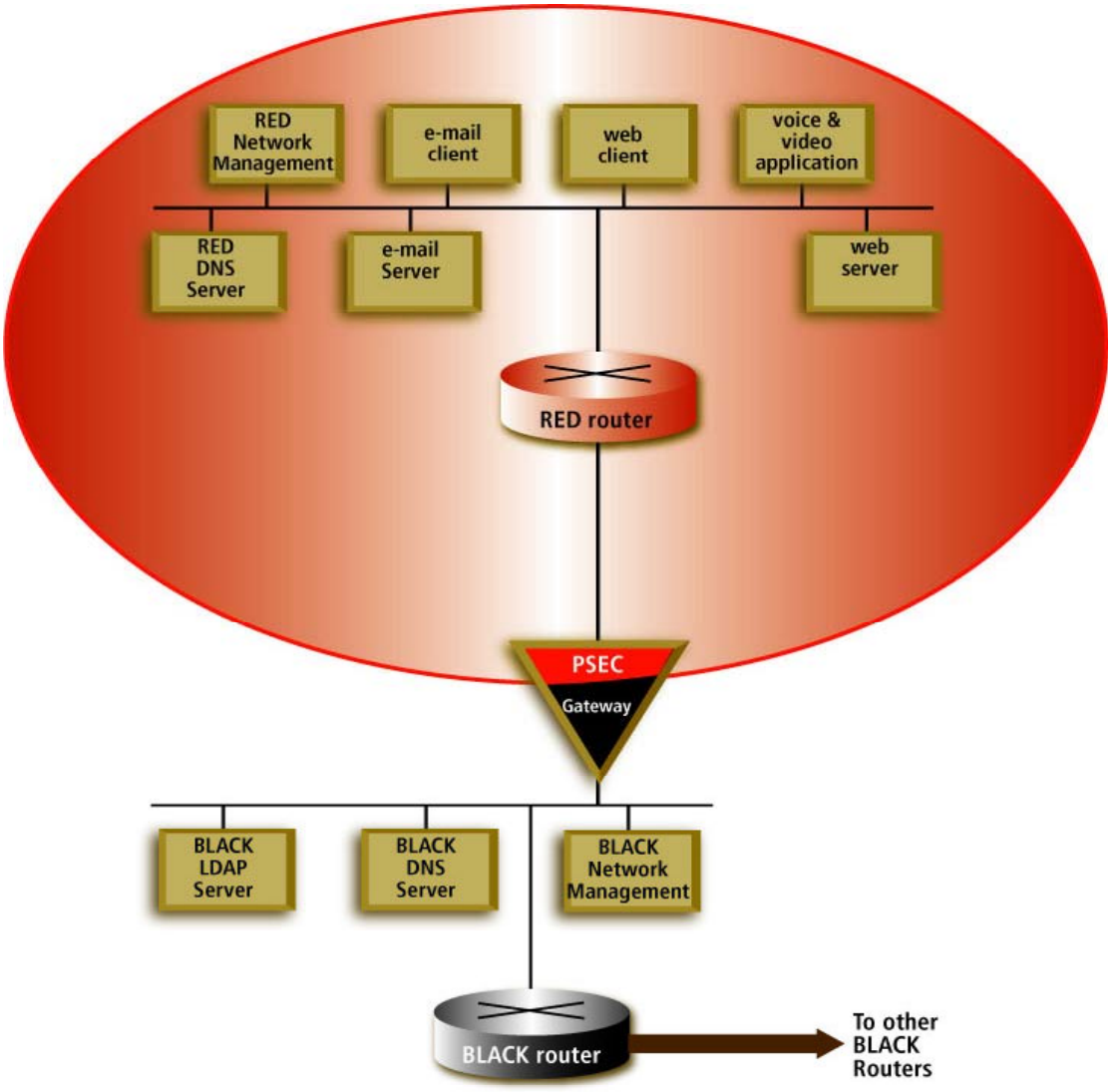

Figure 4. *Coalition LAN*

2.5 The Canadian INSC Testbed

Canada implemented an INSC testbed that connected to all three wide area networks and included the following four CLANs, shown schematically in Figure 5:

- one CLAN on the JCWAN to support the JCHQ,
- two CLANs on the LWAN to support land operations, and
- one ship CLAN on the MWAN to support maritime operations.

Figure 5 also shows the domain names for these CLANS.

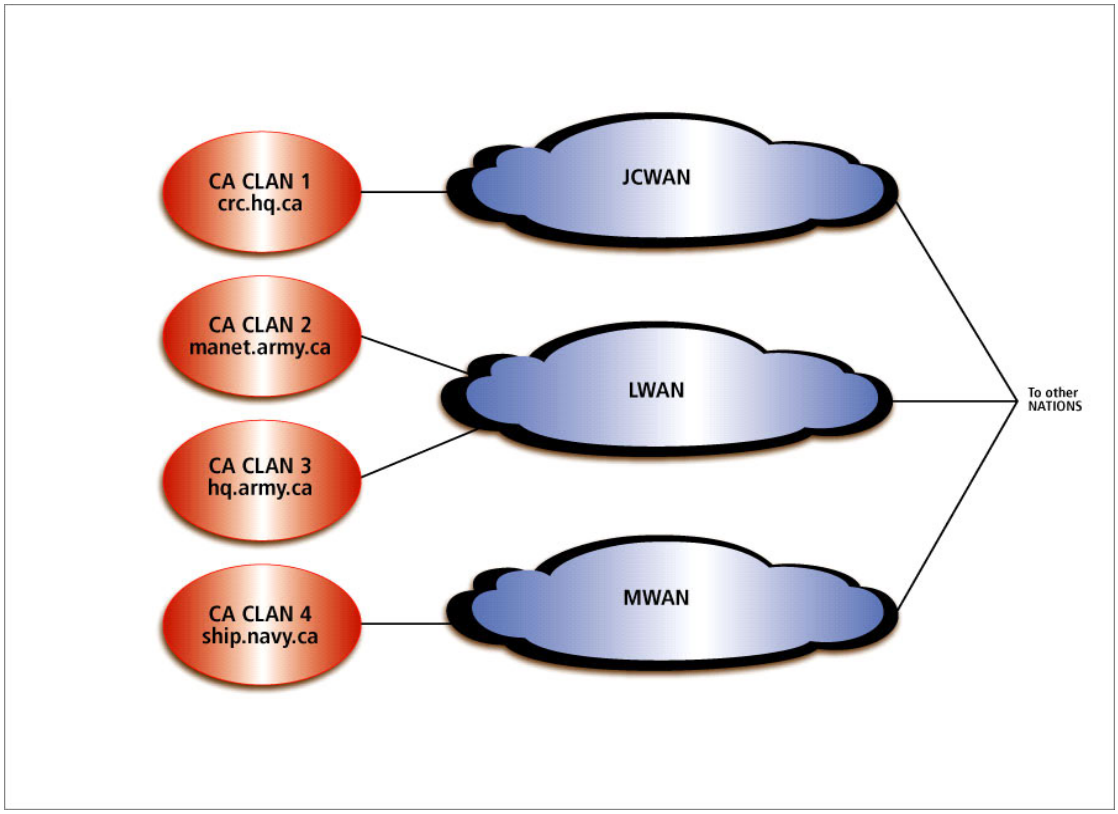

Figure 5. *Canadian CLANS*

For completeness, Figure 6 shows the detailed physical structure of the Canadian INSC testbed including all routers, IPsec devices, end systems, and network links.

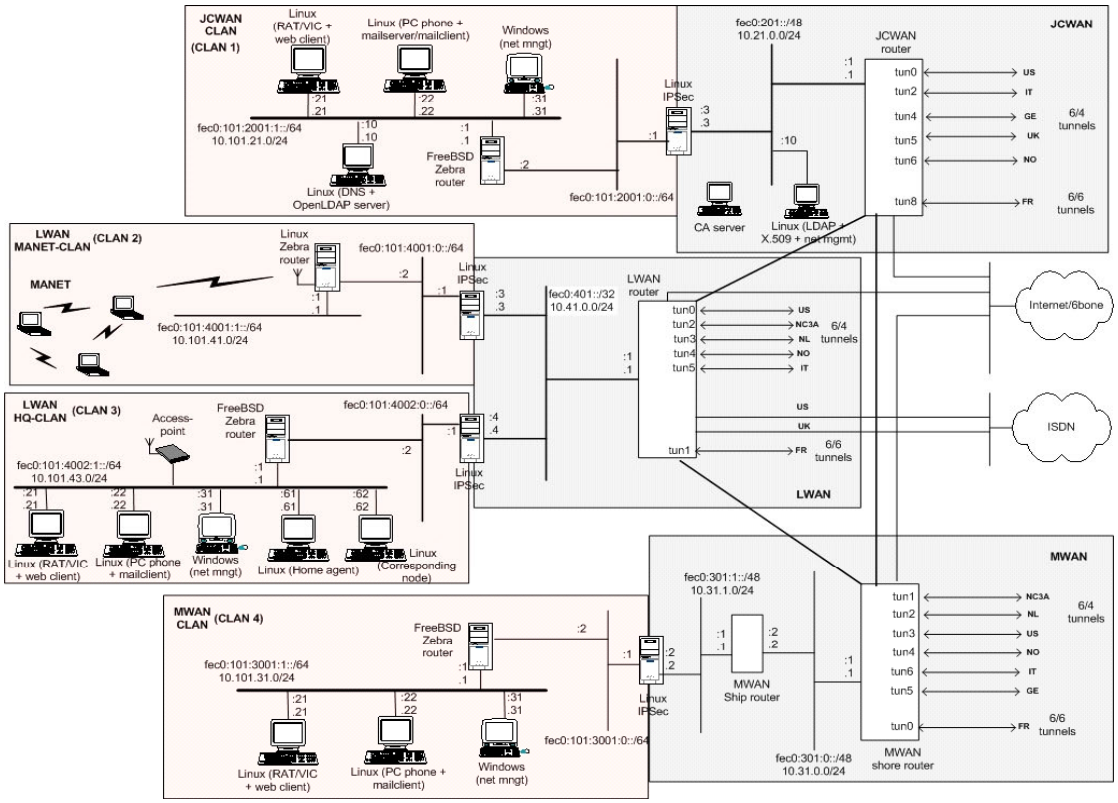

Figure 6. *Canadian INSC Testbed*

3. INSC Security

3.1 Architecture

The INSC security architecture embodied the design principles stated in Section 2.1. The architecture supported a single coalition with two coalition security domains:

- a “black” unclassified domain that contained the wide-area network services (JCWAN, LWAN and MWAN) and
- a “red” protected coalition domain that was comprised of the CLANs.

The black and red domains were separated at the Network Layer by the Internet Protocol Security (IPsec).

Each CLAN or national LAN participating in the coalition connected to the black domain wide-area network services through an IPsec boundary device, also known as an IPsec gateway.

Coalition communications between users and applications in different CLANs took place over virtual private network (VPN) connections tunnelled through the black domain transit networks, which may have included national security domains as well as unclassified public domains. IP packets in the VPN tunnels were protected by IPsec encryption.

All CLANS were assumed to be at the same level of security classification, and it was assumed that there were no JC, L, or M caveats, so all CLANs belonged to the same VPN.

Figure 7 shows the essence of the security architecture, ignoring any internal structure in the intervening unprotected network.

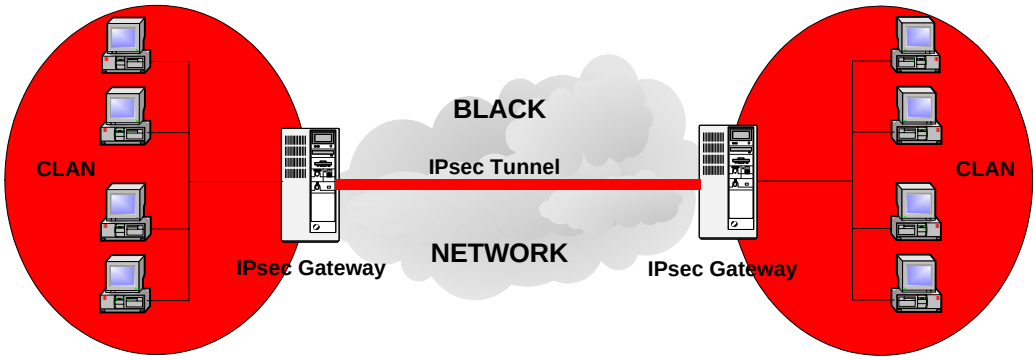

Figure 7. *INSC Security Architecture*

3.2 Scope

The INSC network provided security only at the Network Layer using IPsec. It did not provide any security in the Application Layer.

Figure 8 illustrates several different operational scenarios for which INSC can provide secure communications.

Two CLANs supporting land operations connected to the LWAN through different national tactical networks could establish a secure communication channel through these national networks and the LWAN (solid green line). Similarly, two CLANs aboard ships from different nations could communicate securely across the MWAN (solid green line). In each of these cases, coalition communications were secured in the wide-area networks and within national networks.

Figure 8 also shows how two nationals of the same country, supporting the coalition in different locations, could use a combination of national and INSC assets to establish a secure communication channel for national traffic. In that figure, a soldier in the land forces of Nation 1 supporting the coalition on a national LAN has established a secure communications channel to a sailor aboard a ship of Nation 1, also supporting the coalition (solid red line). In this case the communication path securely traverses the national tactical networks, the coalition wide-area networks and possibly a public network, such as the Internet.

Finally, Figure 8 shows the possibility of a secure coalition subnetwork using a direct link between a headquarters CLAN on the JCWAN and a land operation CLAN on the LWAN (broken green line).

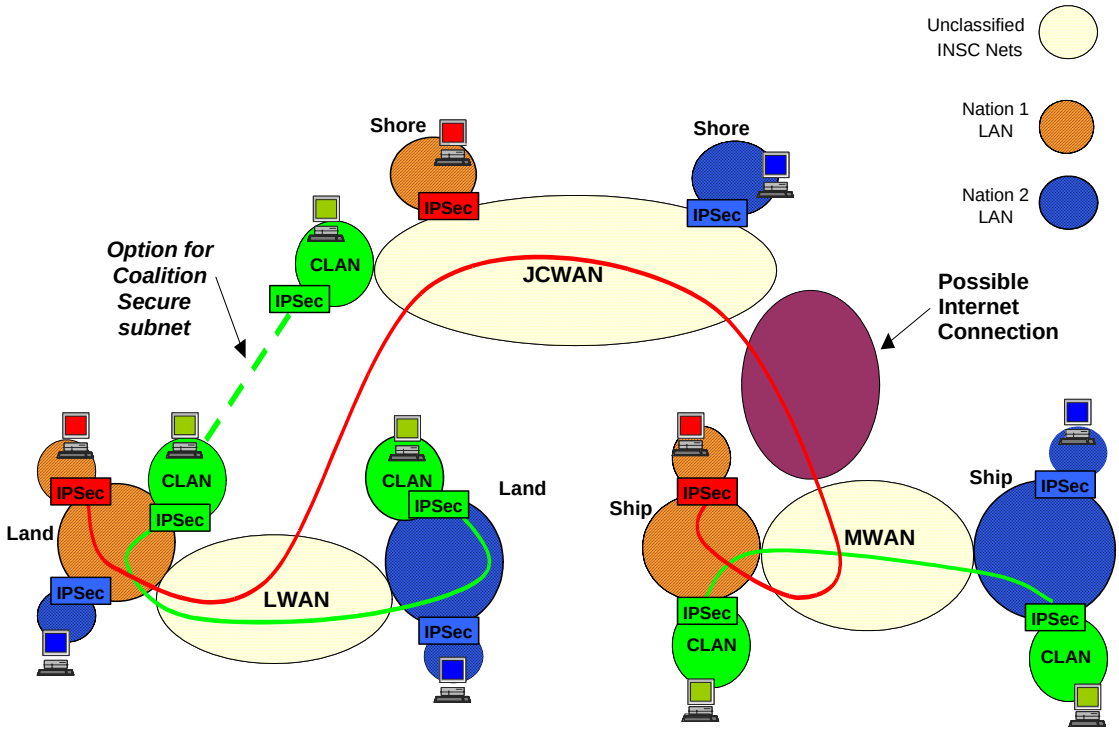

Figure 8. *Scope of INSC Security*

3.3 IPsec Devices

The four Canadian IPsec gateway devices implemented the IABG IPv6 version of FreeS/WAN Linux IPsec on a Red Hat Linux system platform. This IPsec implementation also included code from StrongSec GmbH that enabled FreeS/WAN to use X.509 certificates for device authentication during the negotiation phase to establish a security association (SA).

Other nations used 6WindGate and KAME IPSec implementations on the Berkeley Software Distribution (BSD) Unix platforms (FreeBSD and NetBSD).

Full details of the IPsec implementations used in INSC can be found in the INSC System Configuration [4] and the INSC Task 4 Final Report [10].

4. Security Management

4.1 Purpose

The purpose of a security infrastructure is to implement security mechanisms that provide security services to end users and applications. Once operational, the infrastructure must be managed in accordance with an established security policy in order to maintain the desired level of trust in the security services.

4.2 Requirements

The management of the security infrastructure requires the following elements:

- an overall security policy;
- key management for cryptographic services; and
- administrative and operating procedures for security services and devices.

4.3 Security Policy

A detailed discussion of security policy is beyond the scope of this report. It is sufficient to state that the security policy is essential for managing a security infrastructure. The security policy describes the essential elements of a security posture thereby providing the foundation for trust in the system governed by that security policy.

A typical security policy for a network environment, such as that of INSC, must specify at least the following information:

- The assets to be protected.
- The threats against which protection is required.
- The security services and mechanisms and the protection they provide.
- The operating and administrative procedures for the security services and mechanisms.
- The management roles and responsibilities.
- The procedures and responsibilities assigned to the users.

This list is not intended to be complete and exhaustive, but merely representative.

4.4 Key Management

Security services based on cryptographic mechanisms require the use of cryptographic keys, which must be managed in a secure and trusted manner to maintain the level(s) of trust required by the security policy. Furthermore, it is highly desirable that the key management procedures and mechanisms satisfy a scalability property so that they continue to operate effectively in a dynamic and growing environment.

There are two general approaches to cryptographic key management in a network environment: manual and electronic.

4.4.1 Manual Key Management

Manual key management relies on security administrators to distribute and manage the cryptographic key material through manual procedures. This usually involves the physical delivery and manual loading of key material by security administrators. It calls to mind the image of a courier, carrying the key material in a brief case handcuffed to a wrist, who must travel to each site and deliver the key material to a local security administrator or load the key material into the appropriate devices himself. This method works well in an environment with the following characteristics:

- The community of users is small and/or the environment is limited physically or geographically so that the key material can be distributed effectively in a timely manner.

- The environment is relatively static so that the requirements for key distribution, renewal and revocation are minimized.

This method of key management is secure and trusted, but its main disadvantage is that it lacks scalability. It can quickly become ineffective when the environment becomes large and widely dispersed physically, and the user membership changes frequently so that time required to distribute new key material takes longer than is permitted by the security policy.

4.4.2 Electronic Key Management

Electronic key management, in contrast relies on electronic mechanisms to issue, distribute and manage the key material. This approach is scalable, so it is effective for all sizes of communities, especially larger ones, for any geographical or physical distribution of the environment, and for dynamic environments. The main disadvantage in using electronic management is that many commercial off-the-shelf (COTS) implementations lack formal evaluation and certification by national security agencies and do not, therefore, provide the level of trust required for higher assurance environments.

4.5 Administrative and Operating Procedures

Security policy affects the operating and administrative procedures for the security services and mechanisms because all the procedures must comply with the policy. For the IPsec devices, the policy determines the conditions and procedures for loading, using and managing the device key material. In the case of security services, the security policy determines the conditions and methods governing how and when users and applications use the security services.

5. Electronic Key Management (EKM)

This chapter describes the rationale for a public key infrastructure (PKI) approach to electronic key management and the components of a PKI. Chapter 6 describes how INSC implemented these elements.

5.1 Rationale for Public Key Infrastructure (PKI)

Choosing a key management system in a particular case depends on many factors, but is influenced significantly by the availability of technology to provide electronic management and by interoperability requirements. Manual key management is the default option for many security devices, such as link or network encryption devices, and some device designs may even preclude the use of many electronic key management mechanisms. However, as discussed in Section 4.2, manual key management may not scale, so that electronic key management becomes the only viable alternative.

Furthermore, if they key management is to be provided using COTS products, and if interoperability of the security services based on open public standards is required, then the only currently available technology solution for electronic key management is a PKI system based on accepted standards.

Proprietary systems were considered and then excluded, since in an international environment, such as INSC, it is not possible to mandate the use of specific proprietary products. In addition, proprietary service interfaces can require the customization of the applications and devices using the services, creating a closed system affecting not only interoperability with external systems, but also system life-cycle maintenance, upgrading, and the ability to adapt to new technologies and applications.

The following sections describe the main components of a PKI system.

5.2 PKI Components

A PKI is an electronic key management infrastructure that provides cryptographic security services and automated life-cycle key management services including the following:

- key generation,
- certificate enrolment, and
- certificate revocation and renewal.

The following sub-sections describe the main components of a PKI.

5.2.1 Security Services

The purpose a PKI is to provide and manage security services using mechanisms based on public key cryptography. The security services include:

- authentication,
- integrity,
- confidentiality, and
- non-repudiation.

5.2.2 Policy Management Authority

The Policy Management Authority (PMA) is the top level PKI authority responsible for specifying and approving all policies and policy-related issues in the PKI. The PMA's primary role is to manage the certificate policies and certification practices statements. The PMA is also the point of contact for cross-certification agreements with other PKIs. When implementing a PKI the PMA is normally established first and remains in place throughout the operation lifetime of the PKI.

5.2.3 Security Policy

As with any security infrastructure, a PKI requires a security policy governing the operation of the PKI and the use of its security services. Such a security policy is necessarily a subset of the overall security policy for the complete security infrastructure. It must comply with, and support the requirements and objectives of the overall security policy.

The goal of the PKI security policy is to establish and maintain the specified level(s) of trust in the security services it supports. A PKI security policy does this by addressing the following subjects:

- Certificate Profiles,
- Certificate Policies (CPs),
- a Certification Practices Statement (CPS), and
- Administrative and operating procedures related to the PKI services and mechanisms.

5.2.4 Certification Authority (CA)

The certification authority (CA) is the anchor point for trust in the PKI services. If the PKI involves a hierarchy of CAs, then the trust is anchored in the CA at the top of the hierarchy, known as the root CA.

The CA's function is to maintain and guarantee the validity of public key certificates within its scope of authority. It accomplishes this by providing the following services in accordance with the certificate policies and certification practices statement:

- signing and publishing the public key certificates issued to lower level CAs and/or users, which may be either human users, application processes, or devices,
- revoking expired or compromised certificates,
- renewing expired certificates that are otherwise valid.

5.2.5 Registration Authority (RA)

The registration authority (RA) is a subordinate function of the CA that handles many of the procedures associated with certificate enrolment such as verification of a user's identity. For a small PKI (i.e., a small user community) the CA may provide these functions itself. However, when the user community is large and widely distributed, one or more RA's will often be used to support the CA.

5.2.6 Directory Server

The directory is the public repository used to store and distribute the PKI users' public key certificates and the certificate revocation lists issued by the CA. It is a critical PKI component. Every use of a PKI service (signing, encryption, decryption, signature validation) involves one or more directory accesses to verify that the certificates involved are valid. PKI implementations based on open standards usually require that the directory service support the Lightweight Directory Access Protocol (LDAP) standards.

5.2.7 Client Software

PKI services are provided to users through a service interface. PKI client software implements this service interface and includes bindings that enable applications to request and receive PKI services. The applications using the services must, of course, also be able to use the bindings to the PKI service interface. Such applications are often described as being "PKI enabled". As an example, a PKI-enabled email client is required in order to request the

digital signature and encryption services that allow it to sign, encrypt and decrypt email messages and documents. Normally a PKI product vendor includes this client software with the PKI, but if the PKI service interface conforms to open standards, it may be possible to use third-party and even custom-developed client software.

5.2.8 Administrative and Operating Procedures

Trust in the security of PKI services is based on compliance with the PKI security policy. However, this policy is usually expressed in a form that is independent of the particular implementation. Administrative and operating procedures address the implementation-dependent aspects of compliance with the security policy. It is usually these procedures that change as the implementation aspects of the environment change, typically due to technology evolution or changing user requirements, while the security policy remains. An example of such procedures is the instructions specifying how particular security devices are to be initialized.

Although these procedures are an essential part of an operational PKI, the PKI product vendors cannot provide these procedures. They must be established by the administrative authorities responsible for the operation of the PKI services.

6. The INSC Key Management Infrastructure (KMI)

The purpose of the INSC KMI was to provide secure key distribution and management for the IPsec devices in the INSC network. This chapter describes the details of the key management infrastructure that was implemented for INSC.

6.1 Scope of INSC KMI

The INSC KMI was implemented in the black (unprotected) network domain. The purpose of this KMI was to support an X.509-based [11] authentication service that the IPsec devices could use in the process of negotiating and establishing a security association. There was no need for KMI services in the red domain.

6.2 INSC Security Policy

INSC did not develop a formal security policy or a key management security policy.

An INSC Security Policy Framework document [12] identified a number of security policy issues that an INSC security policy should address. These included the following:

- Identification and classification of security domains,
- Cryptographic Algorithms,
- Key management,
- Device identification,
- Routing issues,
- Security mechanisms and layer placement, and
- IPsec profiles.

An INSC Security Policy document [13] was issued later in the project but its scope was limited and it did not address key management. It did little more than summarize the basic architectural principles of INSC security, such as the following:

- There is only one coalition.
- There are two coalition security domains, a red, or protected, coalition domain and black, or unprotected domain.
- All coalition LANs belong to the red domain and operate at the same level of classification.

- The red coalition domain is separated from the black domain and protected by IPsec security mechanisms.
- Security services are provided only at the network layer using the IPsec mechanism.
- There are no Application Layer security services or mechanisms.
- All INSC data is UNCLASSIFIED.

6.2.1 Trust Model

As discussed in Section 5.2.4, the root CA is the anchor of trust in the PKI security services. When more than one CA is deployed and these CAs operate as peers, cross-certification agreements are usually established so that the users in one CA domain can trust the certificates signed by one of the other CAs. A cross-certification agreement between two CAs involves having each (root) CA sign the other CA's certificate.

Ultimately, at least five INSC nations deployed a CA. In lieu of formal cross-certification agreements, it was agreed that each INSC nation would trust the certificates signed by the CA of any other INSC nation. This was deemed to be sufficient for INSC because it was a closed environment, only unclassified data was involved, and the goal was to investigate and demonstrate the operation of the technology.

With this trust model, each nation operating a CA circulated its root CA certificate to all of the other INSC nations using an out-of-band mechanism. The most common mechanism used an email attachment sent over the Internet.

6.2.2 Certificate Policy

INSC did not develop a certificate policy (CP). Since the goal of INSC was to investigate and demonstrate the operation and use of the technology, developing a formal CP was not a priority.

6.2.3 Certification Practices Statement

INSC did not develop a formal Certification Practices Statement (CPS). A formal CPS was not a priority for the same reason as stated in 6.2.2 for the CP.

6.2.4 Certificate Profiles

Minimal profiles were defined and agreed in the INSC project for root CA certificates, device certificates, certificate revocation lists (CRLs) and authority revocation lists (ARLs). These profiles, which contained the minimum information needed for interoperability, are provided in Annex A.

6.3 Certification Authority

Since the INSC network implemented the IPv6 protocol, the main problem was to find a CA implementation that operated over IPv6. A survey of leading commercial PKI products revealed that none of those products could operate over an IPv6 network. Further investigation determined that it was possible to implement a CA using OpenSSL software on a Linux system platform. This solution was adopted for INSC because the OpenSSL, being open source software, was available to all INSC nations at no cost. Five nations (Canada, Germany, France, the UK, and the US) implemented an OpenSSL CA.

Although the CA could operate over an IPv6 network Canada decided to implement its CA off-line, as a standalone system disconnected from the INSC network, for two reasons. An off-line CA was more secure since it could not be subjected to an attack or compromise from the network. It was also simpler to implement an off-line CA because an on-line CA would have required the development and implementation of an interface to support the CA interaction with client IPsec devices.

The Canadian CA was hosted on an Intel PC running a Red Hat Linux 7.3 operating system with OpenSSL version 0.9.6g. The CA services were implemented using shell scripts in the */etc/ssl/CA* directory executed manually by a CA administrator. These services and scripts are described in the next chapter. All information transferred between the CA, the IPsec devices and the local LDAP directory was done manually using diskettes.

Since the emphasis in INSC was technology investigation, no extraordinary security mechanisms were implemented to protect the CA system. A username and password combination known only to the administrator was used for the administrator account, and no user accounts were configured on the system.

6.4 Directory Service

The purpose of the black domain directory service was to distribute and store the IPsec device certificates and the CRLs. As with the CA, a survey of commercial products determined that there was no commercial IPv6-capable X.500 or LDAP directory server available, so the open source implementation, OpenLDAP, was used for the INSC directory server. Three nations (CA, DE, and US) implemented OpenLDAP directory servers in the black domain on the JCWAN. Figure 4 shows the placement of the black domain LDAP server with respect to a CLAN on the JCWAN. Figure 9 shows the configuration of the three black domain LDAP servers as implemented in INSC.

A combination of LDAP shell commands, a Java client, and a free version of a commercial IPv6-enabled LDAP client were used for directory administration. Nations that did not operate a black domain LDAP server arranged with a nation operating a server to host their entries.

Certificate and CRL information was transferred between the CA and the local LDAP server manually on diskettes. The directory entries containing the certificate and CRL information were updated by executing shell scripts stored in a local directory on the server.

Each LDAP server replicated all of the information that it mastered to the other two servers so that each server held a complete copy of the INSC directory information base. Replication from a server was triggered by a change to information in the local database. This mechanism maintained synchronization with the other servers. The complete redundancy of information in all servers provided survivability.

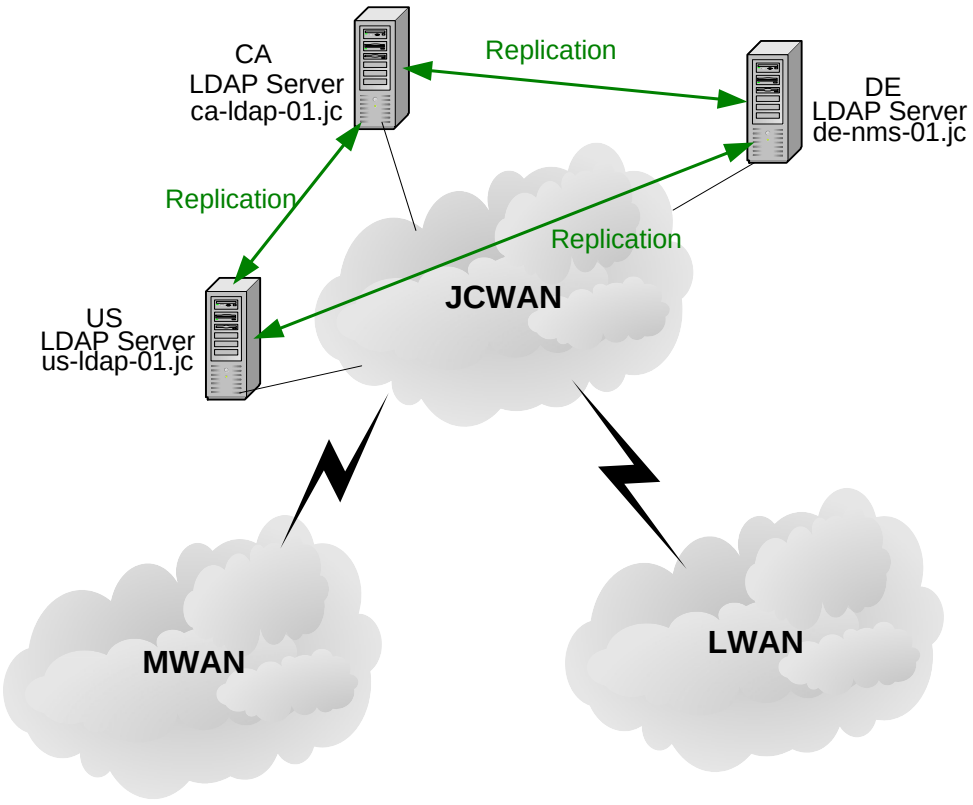

Figure 9. *Black Domain LDAP Configuration*

6.5 Operating Procedures

The configuration and operating procedures implemented to provide key management services for IPsec devices in the Canadian INSC domain are described in Annex B.

7. National Demonstration

7.1 Introduction

Nations in the INSC project held national demonstrations during the period September – November 2003 to showcase the achievements of the INSC project and their national contributions to the project. The Canadian demonstrations, held in November 2003, included presentations on the INSC project, the INSC network, and the security infrastructure, as well as demonstrations of Canadian contributions to the network management, security infrastructure and mobility components of the network. This chapter describes the Canadian security demonstration.

7.2 Demonstration Background

Security in the INSC demonstration was provided at the network layer by IPsec which encrypts IP packets. The encryption and decryption process used cryptographic keys which must be distributed and managed securely. The dynamic coalition environment requires electronic key management – and, therefore, public key infrastructure, or PKI – if we wish to rely on open standards and COTS products.

In INSC, two IPsec devices established a secure channel (i.e., an IPsec tunnel) over which they exchanged encrypted packets by negotiating a security association. The negotiation process began with mutual authentication of the devices desiring to negotiate a security association. In the authentication process, each device sent the other device its identity credentials for verification. In a PKI these credentials consist of some known data encrypted using the sender's private signing key and an X.509 public key certificate containing the sender's public signing key, which is the only key that can decrypt the data packet successfully.

The receiver first verified that the sender's public key certificate was valid as follows:

1. The receiver checked that the CA that signed the certificate was known and trusted.

2. The receiver checked the certificate validity dates to ensure it was within the period of validity.

3. The receiver check the latest CRL issued by the CA to verify that the certificate had not been revoked.

If conditions 1, 2, and 3 were all true, then the receiver used the sender's public key to decrypt the data. If the decryption was successful, then the sender's identity was deemed authentic.

INSC implemented PKI services to support IPsec device authentication. The main PKI components demonstrated in the Canadian INSC domain were:

- an off-line CA to sign and publish device certificates, revoke certificates and issue CRLs,
- a black domain LDAP directory server to store and distribute certificates and CRLs, and
- the client IPsec devices.

7.3 Demonstration Scenario

The goal of the INSC security demonstration was to show how the key management services, and particularly key revocation, could be used to enhance the coalition commander's ability to respond in a timely manner to a change in the operational picture, thus maintaining coalition security.

The demonstration was based on a scenario in which, initially, a fully meshed network of secure IPsec tunnels interconnected all four Canadian CLANs, and an application on the Canadian JCHQ CLAN was transmitting coalition data (simulated in the demonstration by a streaming video broadcast) to a Canadian coalition ship over one of the IPsec tunnels (Figure 10). Note that there must be a CLAN with an IPsec gateway device on board the ship to allow the ship to participate in coalition communications.

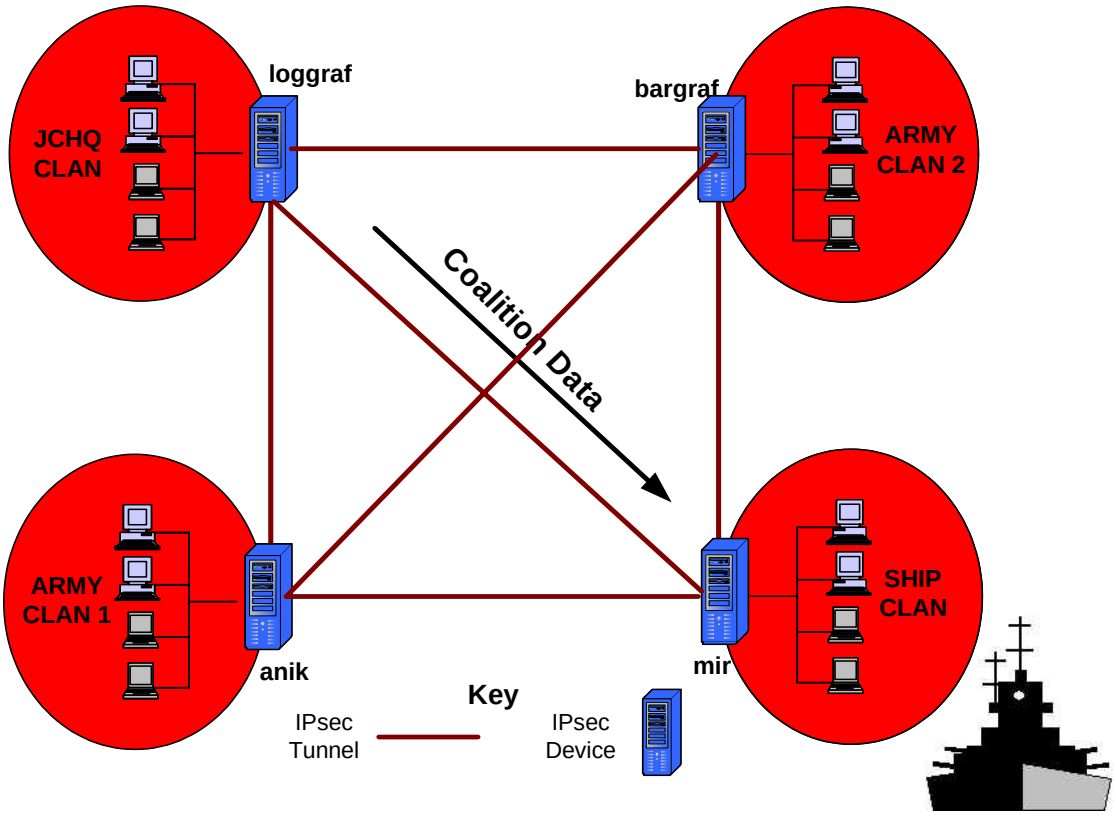

Figure 10. *INSC Demo Scenario: Transmitting Coalition Data*

While coalition information was being transmitted to the ship, it was assumed that an intelligence report reached the coalition commander in the JCHQ that enemy forces had captured the ship (Figure 11). As a result, the commander had to terminate all coalition and national communications traffic to the ship immediately.

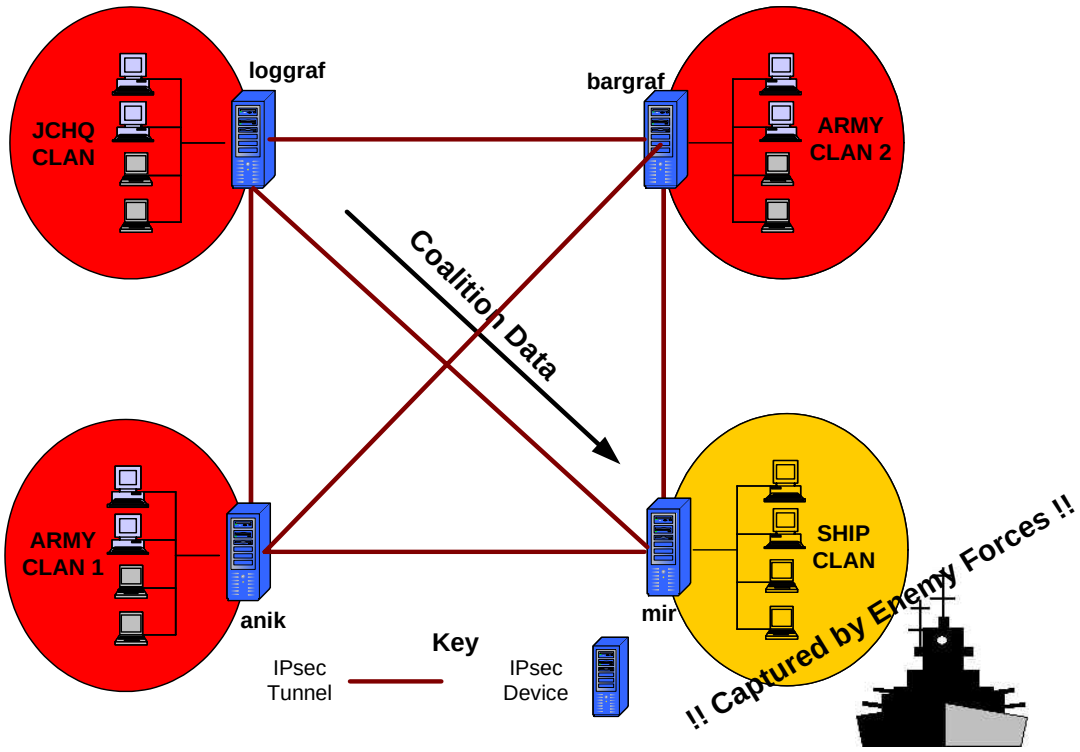

Figure 11. *INSC Demo Scenario: Ship is captured*

The coalition commander immediately ordered the Canadian CA to revoke the ship's IPsec device certificate. The CA revoked the certificate for the ship's IPsec device by issuing a new CRL containing the revoked certificate id and updated the CRL entry in the local LDAP directory with this new CRL (Figure 12).

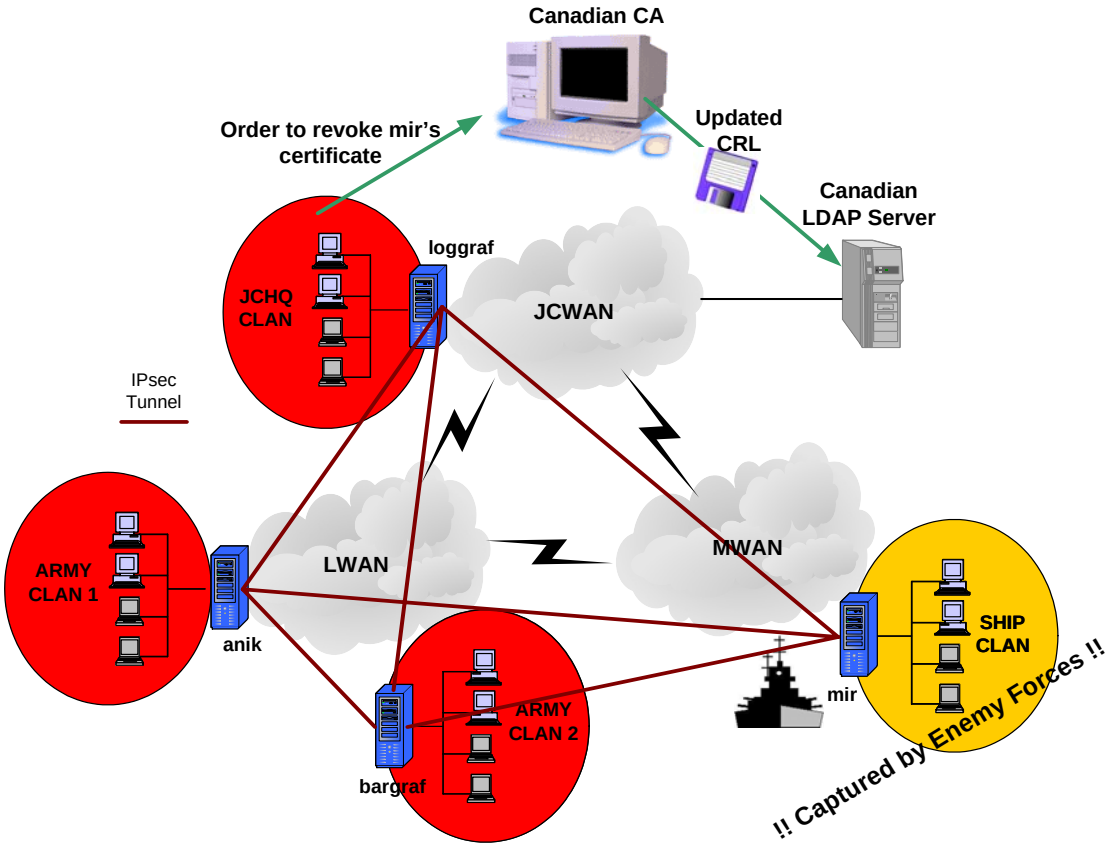

Figure 12. *INSC Demo Scenario: Order certificate revocation*

The CRL update operation in the local LDAP server triggered a replication mechanism, which distributed the new CRL to all remote coalition LDAP servers as well as to all coalition IPsec devices. Immediately on receiving the new CRL, each IPsec device checked the CRL and closed any IPsec tunnel to the captured ship, effectively terminating all communications to the captured ship as soon as the updated CRL was received (Figure 13).

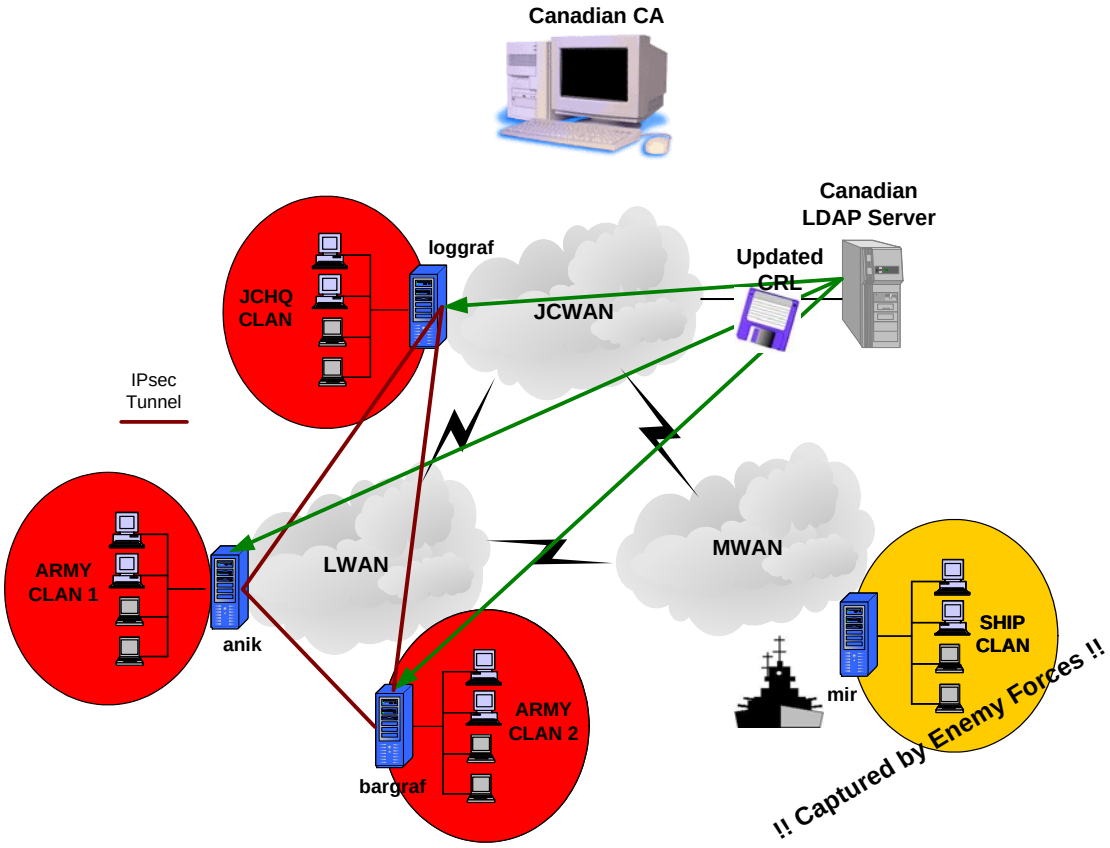

Figure 13. *INSC Demo Scenario: Communications to the captured ship are terminated*

While the coalition commander could have terminated all communications to the captured ship by issuing an order using the normal chain of command to all network and device administrators, the automated CRL mechanism enhanced the commander's ability to respond in a timely manner to changes in battle conditions, especially when the coalition devices are geographically dispersed.

7.4 Implementation Description

The technological capabilities demonstrated by the scenario described in Section 7.3 made use of a combination of existing product capabilities together with custom shell scripts and manual procedures.

The most significant effort was to integrate the IPsec device operation with the LDAP directory and the key management infrastructure. The IABG FreeS/WAN IPsec implementation could use locally stored certificates and CRLs but it had no capability to

query the LDAP directory or to retrieve certificates and CRLs from the directory. Checking locally stored CRLs was activated as described in CRL Checking, Annex B.

The first problem was to implement an automated mechanism to push CRLs to the IPsec devices over the network. The simplest solution that could be implemented quickly with minimal effort was to install an LDAP directory server on each IPsec device with a shell backend and configure LDAP replication from the PKI LDAP server to the server on each IPsec device. When the CRL information was updated on the LDAP directory server using the *update-crl.sh* script, LDAP replication sent the CRL to the LDAP server on each IPsec device where a local script, executed by the shell backend, copied the updated CRL into a local device directory.

The second problem was to force the IPsec device to read the new CRL immediately on receipt and terminate any security association with (i.e., close any IPsec tunnel to) a device whose certificate was revoked by the new CRL. This was achieved by including IPsec device commands to reload the CA certificates and the CRLs and then restart IPsec. The IPsec restart was necessary because, if FreeS/WAN IPsec reloaded the CRL without a restart, it would not terminate existing SAs, even to a device whose certificate had been revoked. The SA would eventually terminate when its lifetime expired, and would not be re-established with a device whose certificate was revoked, but this delay was not acceptable for the demonstration scenario.

8. Observations and Analysis

This section presents observations and lessons learned during the implementation, operation and demonstration of the INSC network.

8.1 IPsec over IPv6

The INSC project found that IPsec/IPv6 is available in both open source and commercial product implementations, although the selection is limited in comparison to the availability of IPsec/IPv4 implementations.

Different IPsec implementations were found to be interoperable, although interoperability was affected by the support for such capabilities as protocol data unit (PDU) fragmentation and reassembly, X.509 certificate-based authentication, and X.509 certificate management. The certificate size was also found to affect interoperability between IPsec devices, in that some devices were able to handle larger certificates than others. This was a significant finding because it could potentially limit the length of the keys used for authentication.

The IPv6 protocol assumes PDU fragmentation and reassembly are handled by the end system so intermediate systems such as routers and gateways do not implement this service. Because the FreeS/WAN IPsec did not perform fragmentation and reassembly, it truncated PDUs that exceeded the maximum size it was configured to handle. Routers also dropped packets that exceeded their maximum PDU size. Since INSC made extensive used of nested tunnels, the resulting PDUs frequently exceeded the maximum size. INSC adopted a pragmatic solution by limiting the maximum transmission unit (MTU) size in all end systems to the same fixed value. This experience highlighted an important shortcoming of the current IPv6 and IPsec standards and implementations.

It was noted that certificates that are too large could also result in IP packets larger than the maximum supported size and thus prevent the successful establishment of SAs. This problem was encountered at least once during testing in Canada. The solution was to reduce the key length, indicating that policy decisions on key length and other aspects of the certificate profile can have performance implications that must be taken into account.

8.2 Key Management Support in IPsec/IPv6

All IPsec/IPv6 implementations supported manual key distribution and management, and authentication using pre-shared secrets.

IPsec/IPv6 support for X.509 authentication was available but was not uniformly implemented, while support for fully integrated electronic key distribution and management was generally not available. The IABG FreeS/WAN implementation supported X.509 authentication, including certificate validation and CRL checking, through a third-party software patch, but it lacked the capability to retrieve certificates and CRLs dynamically from the LDAP directory, so the certificates and CRLs had to be present locally on the IPsec

device. At least one other IPsec implementation in INSC supported X.509 authentication but did not support CRL checking. Commercial products were found to have varying degrees of support for electronic key management ranging from no support to the use of proprietary protocols.

No IPsec/IPv6 implementations were found that implemented application program interfaces for CA and LDAP directory services in order to support complete certificate life cycle management, including automated certificate enrolment, detection of certificate expiry and revocation, certificate renewal, CRL updates, and the dynamic retrieval of certificates and CRLs from an external directory server.

8.3 PKI/IPv6 Support

No commercial PKI products, including X.500 or LDAP directory products were found to operate over IPv6. However, the INSC project did become aware of a prototype PKI/IPv6 implementation from the University of Murcia in Spain, but this discovery was too late to be used in the project.

Consequently, PKI services were implemented using open source CA and LDAP directory software. With this approach it was possible to provide basic certificate enrolment, revocation and renewal services, and it was necessary to use manual operator procedures to transfer data between the CA, the LDAP directory and the IPsec devices. It was not possible to provide complete automated certificate life cycle management largely because the IPsec device did not support the necessary application program interface (API) for CA and directory services.

8.4 Security Policy

Security policy (Section 6.2) was a weak aspect of the INSC infrastructure. No overall security policy was developed in advance to guide the implementation and operation of INSC security although elements of a security policy were adopted in an ad-hoc manner as operational requirements demanded. The sum of these ad-hoc elements did comprise a pragmatic minimal de facto security policy but this served more to describe how the security services were operated rather than to inspire trust in the security services. Nevertheless, even this de facto security policy could be used as the basis of a rigorous security policy.

The obvious lesson is that, in order to provide the desired level of trust in the security services and mechanisms, a security policy and related operating procedures must be planned, developed, and promulgated in advance, before deploying the security infrastructure. Then the implementation and operation of the security services and mechanisms can be shown to conform to the security policy.

It should also be recognized that the development of a rigorous security policy is a major undertaking and, in the case of INSC, would not only have lengthened the project greatly, but would also have detracted from the technology investigation, which was the real focus of the project.

8.5 Trust Model

Although the trust model is an element of security policy, it merits separate mention because it determines the architecture and operation of the KMI. The trust model used in INSC (Section 6.2.1) was adopted as a pragmatic decision because it simplified the operation of the KMI by allowing multiple CAs while avoiding the necessity for cross-certification.

However, the INSC trust model used in the demonstration may also not be the best solution for a coalition operation. One alternative would be to implement a coalition PKI with a root coalition CA administered by the coalition command in accordance with a coalition security policy. An operation involving several disconnected security domains could require multiple subordinate CAs, but they would still be under the authority of the coalition commander. This would obviate the need for cross-certification agreements among the coalition forces of different nations, but cross-certification between the coalition CA and national CAs may still be required to support secure communications between coalition members and their national headquarters.

The analysis of possible trust models and the selection of a suitable candidate to support an international coalition operation was highlighted as an issue but was beyond the scope of the INSC project.

8.6 IPsec Demonstration Scenario

The scenario described in Section 7.3 demonstrated how IPsec with integrated key management functions could enhance the security of communications protected by the IPsec devices. The scenario was based on the operational concept that, if IPsec devices established SAs with remote devices using X.509 certificates for authentication, then those devices should receive automatic notification of revoked certificates and should immediately terminate any SA with a device whose certificate had been revoked. The demonstrated capability involved an on-line dynamic interaction between the LDAP directory and the IPsec devices, such that when a CRL update was published in the LDAP directory, the CRL was immediately communicated to the IPsec devices, which then renegotiated all existing SAs, resulting in the termination of those SAs to any device whose certificate had been revoked.

This capability, implemented for the INSC demonstration using LDAP directory replication, custom shell scripts and manual operating procedures, could not be found in any known commercial or open source IPsec implementation. Furthermore, the INSC demonstration involved an off-line CA, which required a manual procedure to transfer CRLs between the CA and the LDAP directory. Ultimately a fully automated capability would involve an on-line interaction between the CA and the LDAP directory, as well as between the LDAP directory and the IPsec devices.

9. Conclusions and Challenges

9.1 Conclusions

First and foremost, INSC implemented and demonstrated a working security infrastructure for an international IP network. This security infrastructure showed that the use of IPsec devices with electronic key management is a viable approach to secure network communications in an international coalition environment. Although the INSC network focused on the IPv6 protocol, the security infrastructure did not include any inherent IPv6 protocol dependencies and could be used equally in an IPv4 network.

Second, the INSC solution showed that the level of integration of current IPsec/IPv6 implementations with electronic key management is still very immature and full integration remains to be achieved. While the individual components are readily available (IPv6, IPsec, PKI, Directory) a great deal of integration is still needed to provide mature commercial products that possess the capabilities similar to those implemented and demonstrated in INSC. The most important deficiency in this regard is the lack of support for APIs to allow IPsec devices to interact dynamically with PKI services and directory services. There is a need for API implementations based on open standards because proprietary solutions inhibit the ability to adapt and evolve to meet changing requirements and the adoption of new technologies.

Third, INSC has highlighted the need to address the development and management of security policy and security procedures before the deployment of a security infrastructure and to develop and adopt a suitable trust model to support international coalition operations. The INSC results provide a good starting point for further work on trust models and the management of security policy and security procedures.

9.2 Challenges

The highest priority challenge for the future, highlighted by INSC, is to achieve the integration of IPsec with electronic key management and the development of cost-effective commercial products. This is a necessity to provide a scalable system that can be deployed and managed easily in an international coalition environment.

A second challenge is to develop a suitable trust model for an international coalition and a practical security policy for electronic key management.

A third challenge is to extend the IPsec-based security infrastructure to the tactical environment, which includes mobile users, limited communication bandwidths and electromagnetic emission control (EMCON) constraints. This will require substantial progress in both IPsec and PKI implementations and the cooperation of the commercial vendors. Until this is a reality, a scalable manageable security infrastructure will not be possible.

Finally, there is the need to address the evaluation and accreditation of the security infrastructure for operational deployment, which was not included within the scope of the INSC project.

10. References

1. INSC Initial System Architecture, INSC/Task1/D/001, December 2001
2. INSC Test and Demonstration Architecture, INSC/Task1/D/003, June 2003
3. INSC Integration Plan, INSC/Task1/D005, July 2003
4. INSC System Configuration, INSC/Task1/D/008, July 2003
5. INSC Final Report, INSC/Task1/D011, March 2004
6. STANAG 5048 - The Minimum Scale of Connectivity for Communications and Information Systems for NATO Land Forces, Edition 5, 16 Feb. 2000
7. Fundamental Principles for the INSC Demonstration Infrastructure, INSC/Task1/D/006, November 2002
8. Addressing and Naming Plan for INSC Network, INSC/Task1/D/009, March 2003
9. http://www.6bone.net/
10. INSC Task 4 – Security – Final Report, INSC/TASK4/D/006, January 2004-05-04
11. ITU-T RECOMMENDATION X.509 | ISO/IEC 9594-8: *Information Technology – Open Systems Interconnection – The Directory: Public-Key and Attribute Certificate Frameworks*, 4th Edition, 2000
12. INSC Security Policy Framework, INSC/Task 4/D/005, September 2002
13. INSC Security Policy, INSC/Task 4/D/010, July 2003

Annex A: Certificate and CRL Profiles

CA Certificate Profile

1. The CA issuer and subject distinguished name for nation XX, where XX is the 2 character ISO 3166 country code, shall be.

   c=XX, o=INSC, ou=PKI, cn=XX_CA_1

2. It is recommended that CA certificates be valid for a period of time that exceeds the lifetime of the INSC project, such as 10 years.

3. The following X.509 v3 certificate extension fields shall be used:

   Basic Constrains: critical: CA: TRUE
   Subject Key Identifier: non-critical

4. The following X.509 v3 certificate extension fields may be used:

   Authority key identifier: non-critical

User Certificate Profile

1. The issuer distinguished name for nation XX, where XX is the 2 character ISO 3166 country code, shall be:

   c=XX, o=INSC, ou=PKI, cn=XX_CA_1

2. The subject (i.e., IPSec device) distinguished name shall be:

   c=XX, o=INSC, ou=IPsec Devices, cn="device_name",

   where device_name is assigned by the owning nation according to the INSC Naming and Addressing conventions.

3. User certificates should be valid for 2 years from the date of issue.

4. The following X.509 v3 certificate extension fields should be used:

   Authority Key Identifier: non-critical
   Extended Key Usage: non-critical
   Basic Constraints: non-critical: CA: FALSE

5. Additional X.509 v3 certificate extension fields may be used if marked non-critical.

Certificate Revocation List (CRL)

1. CRLs shall be X.509 Version 2.
2. CRL extension fields shall not be used.
3. CRLs shall be stored in the CA directory entry as an attribute.

Authority Revocation List (ARL)

1. ARLs shall be X.509 Version2
2. ARL extension fields shall not be used.
3. ARLs shall be stored in the CA directory entry as an attribute

Annex B: Key Management Operating Procedures

CA Configuration

The following shell scripts (listed in Annex C) were installed in the */etc/SSL/CA* directory of the CA system:

- *init-ca.sh* ,
- *req-cert.sh* ,
- *sign-cert.sh* ,
- *revoke-cert.sh* , *and*
- *generate-crl.sh* .

The CA administrator initialised the CA by executing the script *init-ca.sh* as root. This script removed all files from any old CA, and then generated a CA private key, a self-signed root certificate and an initial signed, but empty, CRL.

The CA root certificate was then distributed to all of the other INSC nations by email.

The CRL was published in the local LDAP directory using the procedure CRL Update in this annex. LDAP replication propagated the CRL to all other directory servers in the INSC domain.

IPsec Device Configuration

Following guidance provided by Germany for FreeS/WAN IPsec, each Canadian IPsec device was configured to use to cryptographic authentication with X.509 certificates as follows:

- the root certificates of all INSC CAs were copied into the local directory */etc/ipsec.d/cacert/* ,
- the CRLs issued by all INSC CAs were copied into the local directory */etc/ipsec.d/crls/* ,
- the shell script *req-cert.sh* was installed in the local */root* directory, and
- the local directory */etc/SSL/keys* was created, owned by the root, with read and write privileges only for the root.

LDAP Directory Configuration

The following shell scripts (listed in Annex D) were installed in the */root* directory of the LDAP directory system:

- *add-entry.sh* , and
- *update-crl.sh* .

Key Generation

The IPsec device administrator used the following procedure to generate the public-private authentication key pair for an IPsec device:

- Log into the root account of the IPsec device and execute the script *req-cert.sh* using the fully qualified domain name (FQDN) of the device as an argument:

  req-cert.sh <fully qualified domain name>

Executing the script creates a private key in the local */etc/SSL/keys* directory, and a certificate request file, containing the public key, on a diskette.

Executing the script *req-cert.sh* locally on the IPsec device ensures that the private authentication key never leaves the device and is therefore only known to the device for which it was created.

Certificate Enrolment

The following procedure, involving the IPsec device administrator, the CA administrator, and the LDAP directory administrator, was used to sign and publish a public key certificate for an IPsec device. In the Canadian INSC domain the same person filled these roles.

1. IPsec device administrator: Transfer the diskette containing the certificate request file generated by the *req-cert.sh* script to the CA.

2. CA administrator: Insert the diskette containing the certificate request file into the CA system and execute the script *sign-cert.sh* with the FQDN of the requesting device as an argument. Executing this script creates the following two files on the diskette:

   - a device certificate based on the information provided in the certificate request file, and
   - a lightweight directory interchange format (LDIF) file to update the LDAP directory

3. IPsec device administrator: Transfer the diskette from the CA to the IPsec device and copy the device certificate into the device directory */etc/ipsec.d/knowncerts/* .

4. LDAP directory administrator: Transfer the diskette to the LDAP directory system and execute the script *add-entry* . This adds an entry to the LDAP directory for the IPsec device that includes the device certificate. Directory replication propagates the change to all other directory servers.

Certificate Revocation

The CA administrator and the LDAP directory administrator used the following procedure to revoke a device certificate.

1. CA administrator: Insert a diskette into the CA system and execute the script *revoke-cert.sh* with the FQDN of the device whose certificate is to be revoked as an argument. Executing this script creates the following two files on the diskette:

    - a CRL signed by the CA that includes the serial number of the revoked certificate, and
    - an LDIF file containing the information required to update the CRL attribute of the CA entry in the directory

2. LDAP directory administrator: Insert the diskette in the directory system and execute the script *update-crl.sh* . This updates the CRL attribute in the CA directory entry with the new CRL. Directory replication propagates the change to all other directory servers.

3. IPsec device administrator: Copy the CRL into the directory */etc/ipsec.d/crls/* of each IPsec device.

CRL Update

The CA administrator and the LDAP directory administrator used the following procedure to update a CRL without revoking a certificate. Normally the policy specifies the CRL lifetime, which is also the frequency of CRL updates, since the CRL must always be valid. Since the Canadian CRL lifetime was one month, this procedure was executed every month to prevent the CRL from expiring.

1. CA administrator: Insert a diskette and execute the script *gen-crl.sh* . Executing this script creates the following two files on the diskette:

    - a CRL signed by the CA, and
    - an LDIF file containing the information needed to update the CRL attribute of the CA entry in the LDAP directory.

2. LDAP directory administrator: Insert the diskette in the directory system and execute the script *update-crl.sh* . This updates the CRL attribute in the CA directory entry

with the new CRL. Directory replication propagates the change to all other directory servers.

3. IPsec device administrator: Copy the CRL into the directory */etc/ipsec.d/crls/* of each IPsec device.

Certificate Expiry and Renewal

There was no automatic renewal process for expired or about-to-expire certificates. The following procedure was followed to renew a device certificate:

1. The existing device certificate was revoked using the Certificate Revocation procedure

2. A new key pair was generated for the device using the Key Generation procedure

3. The new device certificate was enrolled using the Certificate Enrolment procedure

If the CA certificate expired this would invalidate all current certificates signed by the CA with the private key corresponding to the public key in that certificate. The only possible recovery in this case would be to reinitialize the CA using the CA Configuration procedure, revoke all existing device certificates using the Certificate Revocation procedure, and issue new device certificates using the Key Generation and Certificate Enrolment procedures.

CRL Checking

FreeS/WAN IPsec did not check locally stored CRLs automatically by default. To activate the checking of locally stored CRLs it was necessary to include the command

```
strictcrlpolicy=yes
```

in the device configuration file */etc/ipsec.conf*.

Annex C: Certification Authority Scripts

Initialize the CA

```
---------- init-ca.sh ----------
!/bin/sh
##
##  new-root-ca.sh - create the root CA
##  Copyright (c) 2000 Yeak Nai Siew, All Rights Reserved.
##

# Create the master CA key. This should be done once.
if [ ! -f keys/ca.key ]; then
      echo "Removing Old CA Key"
      echo ""
fi

#   make sure environment exists
if [ -d ca.db.certs ]; then
      echo "Removing Old Database"
      rm -r ca.db.certs
fi

if [ -d certs ]; then
      echo "Removing Old Certs"
      rm -r certs
fi

if [ -d keys ]; then
      echo "Removing Old keys"
      rm -r keys
fi

echo "Creating New Database"
mkdir ca.db.certs
mkdir certs
umask 077
mkdir keys
umask 033
echo '01' >ca.db.serial
cp /dev/null ca.db.index

echo "Generating new CA Ket"
umask 077
openssl genrsa -des3 -out keys/ca.key 1024 -rand random-bits
umask 033

# Self-sign it.
CONFIG="root-ca.conf"
```

```
cat >$CONFIG <<EOT
[ req ]
default_bits                    = 1024
default_keyfile                 = keys/ca.key
distinguished_name              = req_distinguished_name
x509_extensions                 = v3_ca
string_mask             = nombstr
req_extensions                  = v3_req
[ req_distinguished_name ]
countryName                       = Country Name (2 letter code)
countryName_default               = CA
countryName_min                   = 2
countryName_max                   = 2
0.organizationName                = Organization Name (eg, company)
0.organizationName_default        = INSC
0.organizationalUnitName            = Organizational Unit Name (eg,
section)
0.organizationalUnitName_default  = PKI
1.organizationalUnitName        = CA Name(eg, XX_CA_1 where XX is the
two digit country code)
[ v3_ca ]
basicConstraints        = critical,CA:true
subjectKeyIdentifier            = hash
[ v3_req ]
nsCertType              = objsign,email,server
EOT

echo "Self-sign the root CA..."
openssl req -new -x509 -days 3650 -config $CONFIG -key keys/ca.key -
out certs/ca.crt

/etc/ssl/CA/gen-crl.sh

rm -f $CONFIG
```

Request a Certificate

```
---------- req-cert.sh ----------

#!/bin/sh
##
##  new-user-cert.sh - create the user cert for personal use.
##  Copyright (c) 2000 Yeak Nai Siew, All Rights Reserved.
##

# Create the key. This should be done once per cert.
CERT=$1
if [ $# -ne 1 ]; then
        echo "Usage: $0 CN"
        exit 1
fi
```

```
if [ -d keys ]; then
      umask 077
      mkdir keys
      umask 033
fi

if [ -d certs ]; then
      mkdir certs
fi

if [ -f keys/$CERT.key ]; then
      echo "Removing old Private Key."
      rm keys/$CERT.key
      echo ""
fi

echo "Generating Private Key: $CERT.key."
umask 077
#
#Add -des3 option to force user to input a protecting passphrase for
the private key
#
/usr/bin/openssl genrsa -out keys/$CERT.key -des3 1024
umask 033
echo ""

CONFIG="cert.conf"
cat >$CONFIG <<EOT
[ req ]
default_bits                    = 1024
default_keyfile                 = keys/$CERT.key
distinguished_name              = req_distinguished_name
string_mask                     = nombstr
req_extensions                  = v3_req
[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = CA
countryName_min                 = 2
countryName_max                 = 2
0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = INSC
0.organizationalUnitName          = Organizational Unit Name (eg,
section)
0.organizationalUnitName_default  = IPsec Devices
commonName                      = Common Name
commonName_default              = $CERT
commonName_max                  = 64
[ v3_req ]
basicConstraints                = CA:false
EOT

/usr/bin/openssl req -new -config $CONFIG -key keys/$CERT.key -out
$CERT.csr
```

```
rm -f $CONFIG

mount /dev/fd0 /mnt/floppy
mv $CERT.csr /mnt/floppy/$CERT.csr
umount /mnt/floppy

echo ""
echo "You may now get it signed.  The Certificate Request file is:
$CERT.csr on the Floppy"
```

Sign a Certificate

```
---------- sign-cert.sh ----------

#!/bin/sh
##
##  sign-cert.sh - sign using our root CA key
##  Copyright (c) 2000 Yeak Nai Siew, All Rights Reserved.
##

CERT=$1
mount /dev/fd0 /mnt/floppy
if [ $# -ne 1 ]; then
        echo "Usage: $0 CN"
        exit 1
fi
if [ ! -f /mnt/floppy/$CERT.csr ]; then
        echo "No /mnt/floppy/$CERT.csr found."
      exit 1
fi
# Check for root CA key
if [ ! -f keys/ca.key -o ! -f certs/ca.crt ]; then
      echo "You must have root CA key generated first."
      exit 1
fi

# Sign it with our CA key #

#   make sure environment exists
if [ ! -d ca.db.certs ]; then
      echo "CA Database Missing."
#    mkdir ca.db.certs
fi
if [ ! -f ca.db.serial ]; then
      echo "CA Serial Number Index Missing."
#    echo '01' >ca.db.serial
fi
if [ ! -f ca.db.index ]; then
      echo "CA Index missing."
#    cp /dev/null ca.db.index
fi
```

```
#  create the CA requirement to sign the cert
cat >ca.config <<EOT
[ ca ]
default_ca              = default_CA
[ default_CA ]
dir                     = .
certs                   = \$dir
new_certs_dir           = \$dir/ca.db.certs
database                = \$dir/ca.db.index
serial                  = \$dir/ca.db.serial
RANDFILE                = \$dir/random-bits
certificate             = \$dir/certs/ca.crt
private_key             = \$dir/keys/ca.key
default_days            = 730
default_crl_days        = 30
default_md              = md5
preserve                = no
x509_extensions         = server_cert
policy                  = policy_anything
[ policy_anything ]
countryName             = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
[ server_cert ]
#subjectKeyIdentifier   = hash
authorityKeyIdentifier  = keyid:always
extendedKeyUsage  = serverAuth,clientAuth,msSGC,nsSGC
basicConstraints  = CA:false
EOT

#  sign the certificate
echo "CA signing: $CERT.csr -> $CERT.crt:"
/usr/bin/openssl ca -config ca.config -out certs/$CERT.crt -infiles
/mnt/floppy/$CERT.csr
echo "CA verifying: $CERT.crt <-> CA cert"
/usr/bin/openssl verify -CAfile certs/ca.crt certs/$CERT.crt
/usr/bin/openssl x509 -inform pem -in certs/$CERT.crt -outform der -
out /mnt/floppy/$CERT.crt

openssl x509 -subject -noout -inform der -in /mnt/floppy/$CERT.crt >
test.file
awk -F '/' '{for (i = NF; i>=2;i--) printf ("%s,"), $i}' < test.file
> test2.file
echo "dn: `sed s/,$// < test2.file`" > /mnt/floppy/add.ldif
echo "objectClass: top" >> /mnt/floppy/add.ldif
echo "objectClass: device" >> /mnt/floppy/add.ldif
echo "objectClass: strongAuthenticationUser" >> /mnt/floppy/add.ldif
echo "`sed "s/=/: /" < test2.file |sed "s/,.*$//"`" >>
/mnt/floppy/add.ldif
echo "userCertificate;binary:< file:///mnt/floppy/$CERT.crt" >>
/mnt/floppy/add.ldif
```

```
#  cleanup after SSLeay
rm /mnt/floppy/$CERT.csr
rm test.file
rm test2.file
rm -f ca.config
rm -f ca.db.serial.old
rm -f ca.db.index.old

umount /mnt/floppy
```

Generate a CRL

```
---------- gen-crl.sh ----------

#!/bin/sh
##
##  sign-user-cert.sh - sign using our root CA the user cert
##  Copyright (c) 2000 Yeak Nai Siew, All Rights Reserved.
##

# Check for root CA key
if [ ! -f keys/ca.key -o ! -f certs/ca.crt ]; then
      echo "You must have root CA key generated first."
      exit 1
fi

#  create the CA requirement to sign the cert
cat >ca.config <<EOT
[ ca ]
default_ca              = default_CA
[ default_CA ]
dir                     = .
certs                   = \$dir
new_certs_dir           = \$dir/ca.db.certs
database                = \$dir/ca.db.index
serial                  = \$dir/ca.db.serial
RANDFILE                = \$dir/random-bits
certificate             = \$dir/certs/ca.crt
private_key             = \$dir/keys/ca.key
default_days            = 365
default_crl_days        = 30
default_md              = md5
preserve                = yes
x509_extensions         = user_cert
policy                  = policy_anything
[ policy_anything ]
commonName              = supplied
emailAddress            = supplied
[ user_cert ]
subjectAltName          = email:copy
basicConstraints  = critical,CA:false
```

```
authorityKeyIdentifier  = keyid:always
extendedKeyUsage  = clientAuth,emailProtection
EOT

#  Generate the CRL
echo "Generating a CRL into ca.crl"
mount /dev/fd0 /mnt/floppy
openssl ca -config ca.config -gencrl -out ca.crl
openssl crl -text -noout -in ca.crl
openssl crl -inform pem -in ca.crl -outform der -out
/mnt/floppy/ca.crl
openssl x509 -inform pem -in certs/ca.crt -outform der -out
/mnt/floppy/ca.crt

openssl x509 -subject -noout -inform der -in /mnt/floppy/ca.crt >
test.file
awk -F '/' '{for (i = NF; i>=2;i--) printf ("%s,"), $i}' < test.file
> test2.file
echo "dn: `sed s/,$// < test2.file`" > /mnt/floppy/add.ldif
echo "objectclass: organizationalUnit" >> /mnt/floppy/add.ldif
echo "objectClass: pkiCA" >> /mnt/floppy/add.ldif
echo "`sed "s/=/: /" < test2.file |sed "s/,.*$//"`" >>
/mnt/floppy/add.ldif
echo "caCertificate;binary:< file:///mnt/floppy/ca.crt" >>
/mnt/floppy/add.ldif
echo "certificaterevocationlist;binary:< file:///mnt/floppy/ca.crl"
>> /mnt/floppy/add.ldif
umount /mnt/floppy

#  cleanup after SSLeay
rm test.file
rm test2.file
rm -f ca.config
rm -f ca.db.serial.old
rm -f ca.db.index.old
```

Revoke a Certificate

```
---------- revoke-cert.sh ----------

#!/bin/sh
##
##  sign-user-cert.sh - sign using our root CA the user cert
##  Copyright (c) 2000 Yeak Nai Siew, All Rights Reserved.
##

CERT=$1
if [ $# -ne 1 ]; then
        echo "Usage: $0 CN"
        exit 1
fi
```

```
if [ ! -f certs/$CERT.crt ]; then
        echo "No $CERT.crt found."
      exit 1
fi
# Check for root CA key
if [ ! -f keys/ca.key -o ! -f certs/ca.crt ]; then
      echo "You must have root CA key generated first."
      exit 1
fi

# Sign it with our CA key #

#  create the CA requirement to sign the cert
cat >ca.config <<EOT
[ ca ]
default_ca              = default_CA
[ default_CA ]
dir                     = .
certs                   = \$dir
new_certs_dir           = \$dir/ca.db.certs
database                = \$dir/ca.db.index
serial                  = \$dir/ca.db.serial
RANDFILE                = \$dir/random-bits
certificate             = \$dir/certs/ca.crt
private_key             = \$dir/keys/ca.key
default_days            = 365
default_crl_days        = 30
default_md              = md5
preserve                = yes
x509_extensions         = user_cert
policy                  = policy_anything
[ policy_anything ]
commonName              = supplied
emailAddress            = supplied
[ user_cert ]
subjectAltName          = email:copy
basicConstraints  = critical,CA:false
authorityKeyIdentifier  = keyid:always
extendedKeyUsage  = clientAuth,emailProtection
EOT

#  revoke the certificate
echo "CA revoking: $CERT.crt:"
openssl ca -config ca.config -revoke certs/$CERT.crt
/etc/ssl/CA/gen-crl.sh

#  cleanup after SSLeay
rm -f ca.config
rm -f ca.db.serial.old
rm -f ca.db.index.old
```

Annex D: LDAP Directory Scripts

Add an Entry

```
---------- add-entry.sh ----------

#!/bin/sh

mount /mnt/floppy
ldapadd -D $1 -W -x -v -f /mnt/floppy/add.ldif
umount /mnt/floppy
```

Update an Entry

```
---------- update-crl.sh ----------

#!/bin/sh

mount /mnt/floppy
#ldapmodify -D $1 -w k@neCas3 -x -v -f /mnt/floppy/add.ldif
ldapmodify -D "cn=Manager, o=INSC, c=CA" -w k@neCas3 -x -v -f
/mnt/floppy/add.ldif
umount /mnt/floppy
```

List of Symbols and Abbreviations

API	Application Program Interface
ARL	Authority Revocation List
AS	Autonomous System
CA	Certification Authority
CJTF	Combined Joint Task Force
CLAN	Coalition Local Area Network
COTS	Commercial off-the-Shelf
CP	Certificate Policy
CPS	Certification Practices Statement
CRL	Certificate Revocation List
DND	Department of National Defence
DNS	Domain Name Service
EKM	Electronic Key Management
FQDN	Fully Qualified Domain Name
INSC	Interoperable Networks for Secure Communications
IPsec	Internet Security Protocol
IPv6	Internet Protocol, version 6
JCWAN	Joint Command Wide Area Network
JCHQ	Joint Command Headquarters
ISDN	Integrated Services Digital Network
JTF	Joint Task Force

KMI	Key Management Infrastructure
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LDIF	Lightweight Directory Interchange Format
LWAN	Land Wide Area Network
MOU	Memorandum of Understanding
MTU	Maximum Transmission Unit
MWAN	Maritime Wide Area Network
NATO	North Atlantic Treaty Organization
NC3A	NATO Consultation, Command and Control Agency
PDU	Protocol Data Unit
PKI	Public Key Infrastructure
PMA	Policy Management Authority
RA	Registration Authority
SA	Security Association
VPN	Virtual Private Network
WAN	Wide Area Network

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defense R&D Canada - Ottawa
Ottawa, ON K1A 0Z4

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

The INSC Security Infrastructure (U)

4. AUTHORS (Last name, first name, middle initial)

Zeber, S.

5. DATE OF PUBLICATION (month and year of publication of document)

December 2004

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

55

6b. NO. OF REFS (total cited in document)

13

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

NIO section, DRDC Ottawa
3701 Carling Avenue
Ottawa K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

5BF27

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC Ottawa TR 2004-156

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Unlimited

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

The INSC project was an international collaborative research and development activity established under an MOU among eight NATO nations to investigate and demonstrate a secure IPv6 network infrastructure capable of supporting a multi-national military coalition operation. The NATO C3 Agency also accepted an invitation to contribute although they did not sign the MOU. The goal was to demonstrate a network infrastructure that supported security, interoperability, manageability, and mobility. Security was provided at the Network Layer using the IPsec protocol. No security was provided at the Application Layer. The project successfully demonstrated a working security infrastructure with electronic key management to support the operation of IPsec devices. The main lesson resulting from this demonstration was that, although IPsec/IPv6 and electronic key management are both available independently as commercial technologies, they have not yet been fully integrated in a manner that provides a secure, easily manageable, scalable security infrastructure.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Coalition networks
IPv6
Network security
VPN technology
IPsec
PKI
X.509 authentication
IPsec key management

Defence R&D Canada

Canada's leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

www.drdc-rddc.gc.ca